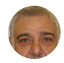
A Post-Classification Scheme for an OCR System for the Notation of the Orthodox Hellenic Byzantine Music

Conference Paper · September 2000

2 authors:

Velissarios Gezerlis
National and Kapodistrian University of Athens



S. Theodoridis
Institute of Electrical and Electronics Engineers

A Post-Classification Scheme for an OCR System for the Notation of the Orthodox Hellenic Byzantine Music

Velissarios G. Gezerlis
University of Athens, Dept. of Informatics,
Div. of Communication and Signal Processing,
Panepistimioupolis, TYPA Buildings,
157 84, Athens, Greece.
e-mail: gbelis@di.uoa.gr

Sergios Theodoridis
University of Athens, Dept. of Informatics,
Div. of Communication and Signal Processing,
Panepistimioupolis, TYPA Buildings,
157 84, Athens, Greece.
e-mail: stheodor@di.uoa.gr

ABSTRACT

In this paper we present, for the first time, the development of a new OCR system for the off-line optical recognition of the characters of the Orthodox Hellenic Byzantine Music Notation, that has been established for use since 1814. We describe the structure of the new system, and propose algorithms for the recognition of the 71 distinct character classes, based on structural and statistical features. For the classification, a tree-structured classification schema and a simple Nearest Neighbor classifier was used. In this paper, an emphasis is given for the description of the post-classification part, which is used in order to increase the overall performance of the system. The final accuracy achieved by the OCR system, tested on a data base of 18,000 characters is 99.3%. The development of such a system is of great importance to musicologists, especially in our days, which are marked by an increased interest, world wide, for the study and understanding of Eastern type musical forms.

1 INTRODUCTION

Byzantine Music is the type of music that was developed, formed and cultured, in the area of the Hellenic Orthodox Christian Church, flourished mainly in the Byzantine epoch, continued its course in the afterwards metabyzantine years and evolved, throughout the years, to the form which is known to day, serving always the warship needs of the Orthodox Church. Byzantine Music originates from the Ancient Hellenic Music that evolved throughout the centuries into the Classical or Western music, developed mainly in western Europe, and into the Hellenic Byzantine Music of our days, as well as the traditional folk music of Hellas [1].

A lot of research effort has been invested over the last forty years, for the development of systems that would be able to understand and optical recognize the western music notes [2]. However, it is the first time that an OCR system is developed for the Byzantine Music (BM).

The goal of this work is to describe a new optical recognition system for the ***Notation of the Hellenic Orthodox Byzantine Music***. The Byzantine Music Notation (BMN), i.e., the way of writing a *psalm*, is of particular interest, not only for the great variety of the used symbols, but also for the way the symbols are combined, making *semantic musical groups* of different meanings.

The OBMR (Optical Byzantine Music Recognition) system is an off-line OCR system and consists of three different and independent stages: a) the ***Segmentation*** stage, b) the ***Recognition*** stage and c) the ***Semantic Musical Group Recognition*** stage. This paper is focused on the Recognition stage and in particular on the Post-Classification part, which has been developed for solving the resulting confusions of BMN characters during the classification stage.

2 CHARACTERISTICS OF THE BYZANTINE MUSIC AND BYZANTINE MUSIC NOTATION

Byzantine Music is a special type of phonetic music, having its own Notation and its own way of performance. An example of the morph and the structure of a *psalm* of the BM is given in fig. 1. As we can notice from this figure, each line of the psalm consists of two parts or one pair. The first one is the Byzantine music Notation (BMN), called *chant.* This consists of about 71 distinct symbols-characters (see table 1), which are combined to form groups of symbols, each one having its own semantic musical meaning and its own musical performance. In the BM more than 2,500 groups of such symbols exist. The second part of each line, corresponding to the singing part of the psalm, consists of the spoken words and syllables written in the Hellenic alphabet [1].

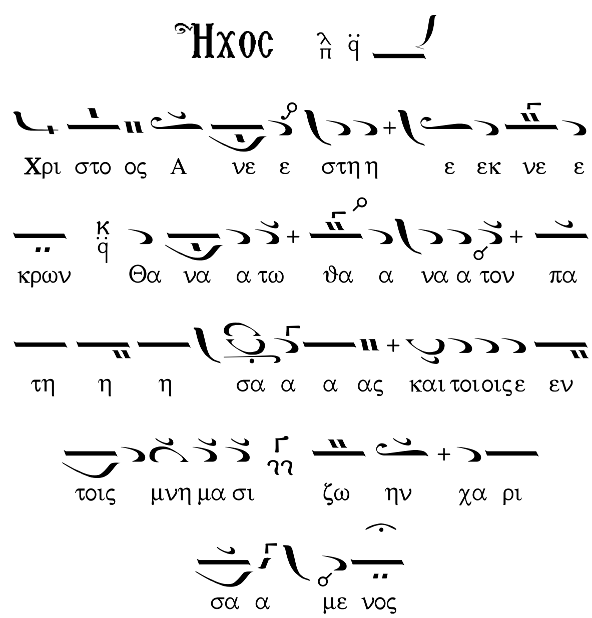


Figure 1. A Byzantine Music psalm called «Christ is Risen».

2.1 The Main Characteristics of the BMN

- The BMN is being written from left to right.
- The BMN characters do not touch each other in a musical text and are always separated (see fig. 1).
- The BMN characters are combined so that to be in left or right, above or below and left or right diagonal position. (fig. 1).
- Many characters of the BMN are identical with others, and are distinguished only by their relative rotation of 45^0, 90^0, 135^0, or 180^0 (see table 1).
- Finally, each one of the resulting semantic musical groups consists of 2 - 10 and even more characters, and corresponds to a different note, or is performed by a *chanter* with a special musical way. This characteristic led us to the conclusion that the OBMR system should extract and recognize semantic musical groups of characters.

2.2 Description of the Data Base of the BMN

For the needs of the system a data base of the BMN was developed, consisted of about 18,000 symbols, in the form of 256 gray scale window bitmap files. We have created 250 different patterns for each one of the 71 classes. These 250 pattern characters consist of: i) real scanned characters from byzantine books ii) printed characters, using a new software for writing the BMN called Byzwriter 1.0 and iii) written by hand carefully so as to resemble to the printed ones. So, the data set is semi-printed and semi-handwritten (see fig. 2).

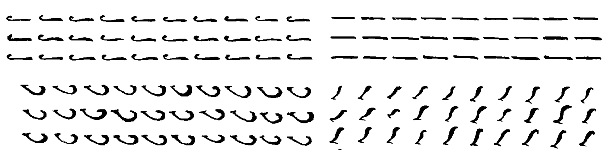


Figure 2. Samples of the BMN data base

3. THE OBMR SYSTEM DESCRIPTION

The overall structure of the OBMR system is divided into the following three stages: a) the ***Segmentation Stage***, which is divided into three tasks: i) to segment the whole page into line pairs, ii) to separate the chant from the Hellenic script below it and iii) to extract the individual characters from the chant. b) The ***Recognition Stage***, which takes as input bitmaps of the characters of BMN, and consists of three main steps: i) the *preprocessing*, ii) the *feature generation*, and iii) the *classification* which will be discussed in detail into the next section. c) The ***Semantic Musical Group Recognition Stage***. It takes as input the character identification numbers (cids), as well as information from the segmentation stage, related to the topological relationship of the characters [9]. Based on this information, we construct the semantic musical groups and recognize them using a data base that works as a grammar and contains all the possible groups of the BMN. Finally, we have the ***post processing*** step, which takes as input the group identification numbers and gives as output the final result which may be the conversion to a true type font, or the performance of the BM.

Table 1. The 71 distinct symbols of the BMN.

isson		fthora Zo		di	
oligon		fthora Ni*		ke	
petasti		fthora Di*		zo	
kentima		fthora Ni**		ni	
ipsili		fthora Pa*		pa	
apostrop.		fthora Di**		vu	
iporoi		fthora Zo*		ga	
elafro		diarkis diesi		delta	
hamili		diarkis ifesi		imifi	
clasma		diesi		lamda	
apli		monogr. diesi		tonos	
varia		digrami diesi		hi	x
stavros	+	trigrami diesi		stigmi	.
komma	,	tetragr. diesi		martiria Ga	
korona		ifesi		martiria Zo	
ifen		monogr. ifesi		martiria Pa	
gorgo		digrami ifesi		spathi	
digorgo		trigrami ifesi		fthora Ni	
trigorgo		tetragr. ifesi		fthora Pa	
argo		syndesm.		fthora Vu	
imiolio		antiken.		fthora Ga	
diargo		endofono		fthora Di	
psifisto		zigos		fthora Ke	
omalo		kliton			

* used for distinguishing the similar names

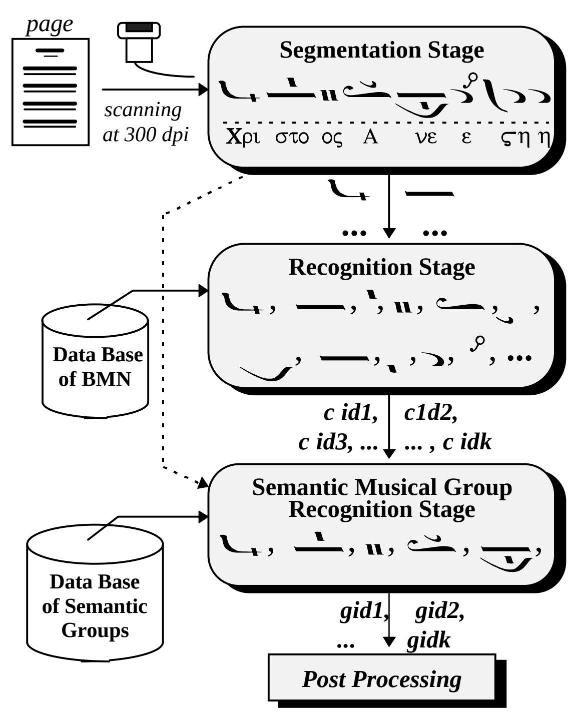


Figure 3. The structure of the OBMR system.

4. THE RECOGNITION STAGE

The three main steps of the recognition stage are the following:

4.1 Preprocessing

The adopted preprocessing algorithms for the image characters are: i) The *Binarization* [4], the *Edge smoothing* for the border of the image character [4], and the *Size normalization* of the image character, fixing the image size to 72x72 pixels.

4.2 Feature Generation

Both structural and statistical features were generated [3], [7]:

4.2.1 Structural features.

i) The *Euler number* (E_N) [4], ii) the *Principal Axis Direction* (PA) [4], and iii) the *Ratio of the Horizontal Bounding Rectangle* (Ratio_of_HBR), of the character is computed. These three structural features constitute an hierarchical preclassification schema that divides the large set of 71 character classes, of table 1, into 20 smaller subsets with an average of 10 characters per subset. This specific hierarchy was the result of extensive experimentation and exploits the specific characteristics of BMN and it leads to an efficient classification system.

4.2.2 Statistical features.

The statistical features that we have used are: a) the Discrete Wavelet Transform (DWT) applied onto the co-ordinate vectors $x(i)$ and $y(i)$, $i=0...127$ [5], of the approximated version of the contour function of the character. For the approximation of the contour the Bezier Spline method was used [6]. b) The DWT applied onto the 4-projection [8] vector $P(i)$, of the character. Specifically, each character is projected in the four directions (horizontal, vertical, left and right diagonal). These projections are then combined to form a single vector $P(i)$, on which the DWT is applied. To our knowledge, this is the first time that such a set of features is used, and was the outcome of extensive experimentation. It turns out that such a feature set exploits the specific characteristics of the BMN symbols.

4.2.3 The Final Feature Vector

For the implementation of the DWT the low and high pass Daubechies «db2» filters were used. The Discrete wavelet decomposition is applied onto the $x(i)$ and $y(i)$, $i=0...127$, vectors of the contour function and onto the $P(i)$, vector of the projections. The final feature vector F is of dimension 60. This consists of the $16 + 16 = 32$ lowest wavelet descriptors of the x and y decomposition plus the 28 lowest wavelet descriptors of the P decomposition [6]. We use this feature vector for the final classification of the characters using a Nearest Neighbor Classifier (NNC) together with the preclassification scheme, described before, which is trained and tested using the described database of the BMN.

5. THE POST-CLASSIFICATION PART

Using the preclassification schema of the structural features as well as the Nearest Neighbor Classifier, the system resulted in an average accuracy of 96,17%. However, we observed from the experiments that some of the characters had low recognition accuracy, which was ranged from 59% to 88%. This was because all these characters were mutually confused with other similar ones.

In figure 4 the following confused characters are presented: a) The symbols «fthora of Vu» and «fthora of Zo» differ only in a vertical small line appearing in the second symbol. b) The symbols «fthora of Ni» and «fthora of Ni*» differ also in a small vertical line present in the second character. c) The three symbols, i.e., «zygos», «fthora of Ni**» and «fthora of Di**» differ only in a diagonal small line. It's obvious that all these are very similar. d) There are four characters which differ in the number of the diagonal lines (i.e., 1, 2, 3, 4 diagonal lines). The same confusion occurs in case of e), where the characters are similar and differ only in the number of the diagonal lines. f) The characters «spathi» and «fthora of Di*» differ only in a vertical small line and g) the characters «apli» and «stigmi», which are too small, resemble both to a dot and the one matches to a square while the other to a small circle.

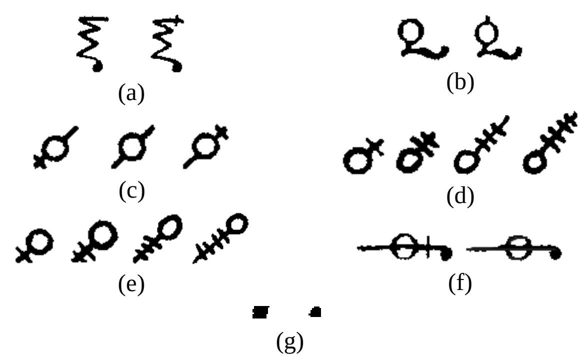


Figure 4: The confusion set of BMN according to the classification schema we described that reduces the total system accuracy. The characters in each subgroup (a - g) are confused each other.

In order to solve these confusions we developed a ***Post-Classification Part***, which consists of one algorithm for each case, based on structural features and exploits the specific structural characteristics of the confused characters. For the development of this post-classification part a large number of «off-the-shelf» techniques was adopted, but none of these gave satisfactory results. A scheme tailored to the specific nature and form of the involved characters had to be developed in order to gain substantially in performance. In the sequel, we present briefly these algorithms:

For the confusion set of fig. 4.a, we developed an algorithm which computes the three points P_1, P_2 and P_3, on the upper part of the two characters (see fig. 4.1), where P_2 is the highest point between P_1 and P_3. Then, we compute the distance d between the point P_2 and the line segment P_1P_3. If $d > T$, where T is a predefined threshold value, then the character is the «fthora of Zo», otherwise is the «fthora of Vu».

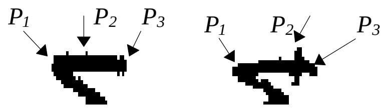


Figure 4.1: The algorithm for the confusion of fig. 5.1.a.

For the confusion set of fig. 4.b, we developed another algorithm which computes the highest point of the upper part of the two characters, P_{max}, and two other equidistant points left and right of P_{max}, P_1 and P_2, shown in fig. 4.2. Then, if the angle θ between the line segments P_1P_{max} and P_2P_{max} is greater than a predefined threshold T, then the character is the «fthora of Ni», otherwise is the «fthora of Ni*».

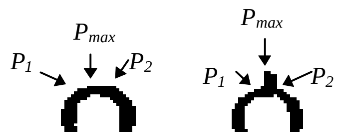


Figure 4.2: The algorithm for the confusion of fig. 4.b.

For the confusion set of fig. 4.c, we developed a different algorithm that estimates the line which connects the right diagonal starting point of the contour, *Sp*, and the contour point *P*, which corresponds to the maximum distance from *Sp*. These two points separate the contour into two parts. The left part is shown in fig. 4.3. A vector C_A is created that contains the distances *d* of the contour pixels of the left part, from the line *SpP* and a vector C_B for the corresponding right part. A smoothing procedure was applied on these vectors several times, using a two by two averaging. It turns out that the resulting smoothed vectors consist of only very small and very large values. Then we search if there are large values in the 8 first and 8 last positions of the vectors. If there are large values in the 8 last positions then we have the character «zygos» (the first one in fig. 4.c). If there are large values in the first 8 positions, then we have the character «fthora of Di**» (the second one in fig. 4.c), otherwise we have the character «fthora of Ni**». This algorithm exploits the symmetric form of the three characters as well as their minor differences.

For the character set of fig. 4.d and 4.e, we used the same algorithm, since these characters are similar and differ only by a relative 180^0 rotation. The algorithm follows the same philosophy of the previous described algorithm (fig 4.3). The only difference is that in this algorithm we search the smoothed vectors C_A and C_B and count the number of maximum and minimum values. Let k_1 be the number of maximum values in C_A and k_2 the corresponding number of minimum values, and also k_3 *and* k_4 are the corresponding minimum and maximum values of the right part C_B. Then we sum up $k = k_1 + k_2 + k_3 + k_4$ and if k is greater or smaller than predefined thresholds, we can separate the four characters of fig. 4.d and the ones of fig. 4.e.

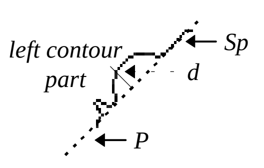


Figure 4.3: The algorithm for the confusion of fig. 4.c. The left part of the contour of character «zygos» and the line between the points *Sp* and *P*.

Concerning the set of fig. 4.f, we used the same algorithm described for those of fig 4.c. Finally, the separation of «apli» and «stigmi» (fig. 4.g) can be achieved easily during the semantic musical group recognition phase, since the two characters are always placed in different semantic positions in the groups.

6. EXPERIMENTAL RESULTS

The Post-classification part, which overcomes the «confusion set» of the fig. 4, together with the structural preclassification schema was the result of extensive experimentation, increasing the overall system accuracy from 96,17% up to 99,3%. The evaluation of the system was carried out via the *Leave-ten-method* (LTM).

Finally, the recognition ability of the system was tested with a number of printed Byzantine musical texts, and the recognition accuracy was ranged from 96% to 100%. Currently, more advanced classifiers, such as Neural Networks and Vector Support machines are investigated.

7. CONCLUSIONS

In this paper an off-line OBMR system for the optical recognition of the BMN was presented, for the first time. The recognition system follows an hierarchical tree-structured philosophy, using structural as well as statistical features together with a post-classification scheme.

8. REFERENCES

[1] D.G.Panagiotopoulos «Theory and Practice of the Church Byzantine Music», Athens 1991.

[2] Bainbridge and N. Carter «Automatic Reading of Music Notation», *Handbook of Character Recognition and Document Image Analysis*, pp. 583-603. 1997.

[3] S. Mori, H. Nishida, H. Yamada, «Optical Character Recognition», *Wiley Series* 1999.

[4] H. Bunke, P.S.P. Wang «Handbook of Character Recognition and Document Image Analysis» *World Scientific*, 1997.

[5] G.C.-H.Chuang, C.-C. Jay Kuo, «Wavelet Descriptors of Planar Curves: Theory and Applications», *IEEE Trans. on Image Processing, vol. 5, no 1,* pp. 56-70, Jan. 1996.

[6] D. Hearn and M.P. Baker «Computer Graphics», *Pentice-Hall*, 1994.

[7] Øivind D. Trier, A. K. Jain, T. Taxt, «Feature Extraction Methods for Character Recognition-A Survey», *Pattern Recognition, Vol. 29, no 4,* pp. 641-662, 1996.

[8] T.H.Hilderbrandt and W. Liu «Optical Recognition of Handwritten Chinese Characters: Advances since 1980», *Pattern Recognition, vol. 26, no. 2,* pp. 205-255, 1993.

[9] S.W.Lu, Y. Ren, C.Y. Suen, «Hierarchical Attributed Graph Representation and Recognition of Handwritten Chinese Characters», *Pattern Recognition, vol. 24, no 7,* pp. 617-632, 1991.